<1> Assume that we make an enhancement to a computer that improves some mode of execution by a factor of 3. Enhanced mode is used 50% of the time, measured as a percentage of the execution time when the enhanced mode is in use. Recall that Amdahl's law depends on the fraction of the original, unenhanced execution time that could make use of enhanced mode. Thus, we cannot directly use this 50% measurement to compute speedup with Amdahl's law. (The result is rounded to two decimal places)

(a) What is the speedup we have obtained from fast mode?

(b) What percentage of the original execution time has been converted to fast mode?

<2> When making changes to optimize part of a processor, it is often the case that speeding up one type of instruction comes at the cost of slowing down something else. For example, if we put in a complicated fast floating-point unit, that takes space, and something might have to be moved farther away from the middle to accommodate it, adding an extra cycle in delay to reach that unit. The basic Amdahl's law equation does not take into account this trade-off. (The result is rounded to two decimal places)

(a) If the new fast floating-point unit speeds up floating-point operations by, on average, 3×, and floating-point operations take 30% of the original program's execution time, what is the overall speedup (ignoring the penalty to any other instructions)?

(b) Now assume that speeding up the floating-point unit slowed down data cache accesses, resulting in a 2× slowdown (or 1/2 speedup). Data cache accesses consume 10% of the execution time. What is the overall speedup now?

(c) After implementing the new floating-point operations, what percentage of execution time is spent on floating-point operations? What percentage is spent on data cache accesses?

<3> Your company has just bought a new Intel Core i5 dual-core processor, and you have been tasked with optimizing your software for this processor. You will run two applications on this dual core, but the resource requirements are not equal. The first application requires 60% of the resources, and the other only 40% of the resources. Assume that when you parallelize a portion of the program, the speedup for that portion is 2. (The result is rounded to two decimal places)

(a) Given that 50% of the first application is parallelizable, how much speedup would you achieve with that application if run in isolation?

(b) Given that 80% of the second application is parallelizable, how much speedup would this application observe if run in isolation?

(c) Given that 50% of the first application is parallelizable, how much overall system speedup would you observe if you parallelized it?

(d) Given that 80% of the second application is parallelizable, how much overall system speedup would you observe if you parallelized it?

<4> When parallelizing an application, the ideal speedup is speeding up by the number of processors. This is limited by two things: percent-age of the application that can be parallelized and the cost of communication. Amdahl's law takes into account the former but not the latter. (The result is rounded to two decimal places)

(a) What is the speedup with N processors if 80% of the application is parallelizable, ignoring

the cost of communication?

(b) What is the speedup with 8 processors if, for every processor added, the communication overhead is 1% of the original execution time.

(c) What is the speedup with 8 processors if, for every time the number of processors is doubled, the communication overhead is increased by 1% of the original execution time?

(d) What is the speedup with N processors if, for every time the number of processors is doubled, the communication overhead is increased by 1% of the original execution time?

(e) Write the general equation that solves this question: What is the number of processors with the highest speedup in an application in which P% of the original execution time is parallelizable, and, for every time the number of processors is doubled, the communication is increased by 1% of the original execution time?